



UNITED STATES PATENT AND TRADEMARK OFFICE

A
UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/917,958	07/30/2001	Eric Lawrence Barsness	ROC920010091US1	9346

7590 08/11/2005

Gero G. McClellan
Thomason, Moser & Patterson, L.L.P.
Suite 1500
3040 Post Oak Boulevard
Houston, TX 77056-6582

EXAMINER

SHRADER, LAWRENCE J

ART UNIT	PAPER NUMBER
----------	--------------

2193

DATE MAILED: 08/11/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/917,958

Applicant(s)

BARSNESS ET AL.

Examiner

Lawrence Shrader

Art Unit

2193

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☐ Responsive to communication(s) filed on 24 June 2005.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-36 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-36 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This office action is in response to the amendment filed by the Applicant on 6/24/2005.
2. Claims 1 – 36 remain rejected.
3. The Applicant's arguments have been fully considered, are not persuasive.

Specification

4. The use of the trademark JAVA has been noted in this application. It should be capitalized wherever it appears and be accompanied by the generic terminology.

Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and every effort made to prevent their use in any manner which might adversely affect their validity as trademarks.

Claim Rejections - 35 USC § 112

5. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

6. Claims 1, 12, 16, 20, and 32 rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter, which was not described in the specification in such a way as to reasonably convey to one skilled in the

Art Unit: 2193

relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

The amended claim language recites that the optimized version of source code “corresponds to an original version of the source code,” but the specification discloses that the optimized source code “represents” the original source code (para. [0042]). A common dictionary meaning of “correspond” is: to be compatible, similar or consistent; coincide in their characteristics. Whereas a common dictionary meaning of “represent” is: To be the equivalent of, or to serve as an example of. “Correspond” has a tighter meaning than “represent.” Therefore, it is not clear what the optimized version of source code actually is. It could be a simple functional representation, or a tighter literal correspondence. Even then, what kind of correspondence would be intended? A functional correspondence? Or an exact reproduction of written codes?

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

8. Claims 1, 3, 4, 6; 20, 23, 24, and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Greyzck et al., U.S. Patent 5, 361, 354 (hereinafter referred to as Greyzck), in view of Brandes, U.S. Patent 5,946,484.

In regard to claim 1:

"generating, from optimized object code, the optimized version of source code, wherein the optimized version of source code corresponds to an original version of the source code, modified to reflect an optimization in the optimized code; and

displaying the optimized version of source code on an output device to visually distinguish changes to the original version of the source code in accordance to a compiler optimization, relative to the optimized version."

Greyzck discloses generating an optimized code from an original source code (e.g., Figure 1; column 3, lines 25 – 29), but does not explicitly disclose displaying the optimized code on an output device to view the changes. However, computer systems with a means for visually examining changes in code in order to determine differences are well known in the art.

Therefore, it would have been obvious to one skilled in the art at the time the invention was made to modify the Greyzck invention with the well-known use of a device, for example a printer or a monitor, in order to examine differences because one would be motivated to verify that modifications have accomplished the intended goals.

Additionally, Greyzck evaluates source code and outputs an optimized source code (column 3, lines 25 – 29), but does not disclose that the change is performed in accordance with a compiler optimization. However, Brandes teaches the well-known procedure of decompilation in which source code is produced from optimized object code (column 1, lines 39 – 50). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the generation of optimized source code as taught by Brandes with source code optimization of Greyzck because the one would be motivated to combine the optimization techniques of Greyzck in a compiling function in order to avoid massive rewrites of old code as taught by Brandes at column 1, lines 9 – 31.

In regard to claim 3, incorporating the rejection of claim 1:

"...wherein the optimized source code comprises a decompiled version of an object code generated from the original source code."

Greyzck discloses the optimization of original source code, but does not disclose generating an optimized source code comprising a decompiled version on an object code generated from the original source code. However, Brandes discloses generating source code from object code (column 1, lines 39 – 50). If the object code is optimized, then the recovered source code would be inherently optimized. Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine method of inputting the source code to be optimized of Greyzck with the source code generator of Brandes, because the combination provides the capability to recover source code to thoroughly analyze the source code in order to rewrite or make changes (e.g., after optimization) as taught by Brandes at column 1, lines 9 – 31.

In regard to claim 4, incorporating the rejection of claim 1:

"...wherein the original source code is compiled according to a compiler optimization comprising at least one of an inlining optimization, a common subexpression elimination, a loop invariant code removal, and a dead code elimination."

Greyzck compiles the optimized code having removed loop invariant expressions (see Abstract), but does not disclose generating an optimized source code wherein the original source code is compiled according to a compiler optimization. However, Brandes discloses generating source code from object code (column 1, lines 39 – 50). If the object code is optimized, then the recovered source code would be inherently optimized. Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine method of inputting the source code to be optimized of Greyzck with the source code generator of Brandes, because the

Art Unit: 2193

combination provides the capability to recover source code to thoroughly analyze the source code in order to rewrite or make changes (e.g., after optimization) as taught by Brandes at column 1, lines 9 – 31.

In regard to claim 6, incorporating the rejection of claim 1:

"...wherein the optimized object code comprises bytecode generated using a Java compiler."

Official notice is taken that using original source code comprising bytecode generated using a JAVA compiler would have been well known in the art at the time the invention was made.

In regard to claim 20 (a computer readable medium), rejected for the same reasons put forth in the rejection of claim 1 (a corresponding method).

In regard to claim 23 (a computer readable medium), incorporating the rejection of claim 20, rejected for the same reasons put forth in the rejection of claim 3 (a corresponding method).

In regard to claim 24 (a computer readable medium), incorporating the rejection of claim 20, rejected for the same reasons put forth in the rejection of claim 4 (a corresponding method).

In regard to claim 26 (a computer readable medium), incorporating the rejection of claim 20, rejected for the same reasons put forth in the rejection of claim 6 (a corresponding method).

9. Claims 8, 9; 12, 14 – 16, 18, 19; 28, 29; 32; 35, and 36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Greyzck et al., U.S. Patent 5, 361, 354, in view of Brandes, U.S. Patent 5,946,484, as applied to claim 1 above, and further in view of Percival et al., U.S. Patent 6,226, 652 (hereinafter referred to as Percival)

Art Unit: 2193

In regard to claim 8, incorporating the rejection of claim 1:

"...wherein the optimized source code and the original source code are simultaneously displayed in separate windows of a user interface on the output device."

Neither Greyzck nor Brandes discloses simultaneously displaying the original source code and optimized source code on an output device to visually indicate a change to the original code. However, Percival discloses a visual display of file information indicating highlighted differences (column 4, lines 3 – 4; Figures 3 – 9) between two files. Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the generation of optimized source code as taught by the combination of Greyzck and Brandes, and further combined with the visual output display taught by Percival, because Percival discloses a way to compare files as taught at column 4, lines 5 – 7 and the combination allows the changed performed to the code to be displayed comparing the difference before and after the optimization.

In regard to claim 9, incorporating the rejection of claim 1:

"...wherein displaying comprises:

identifying a difference between the original source code and the optimized source code, where the difference is due to the compiler optimization used by a compiler program when compiling the original source code to generate the optimized source code; and

highlighting the differences on a user interface on the output device."

Neither Greyzck nor Brandes discloses simultaneously displaying the original source code and optimized source code on an output device to visually indicate a change to the original code. However, Percival discloses a visual display of file information indicating highlighted differences (column 4, lines 3 – 4; Figures 3 – 9) between files. Therefore, it would have been

Art Unit: 2193

obvious to one skilled in the art at the time the invention was made to combine the generation of optimized source code as taught by the combination of Greyzck and Brandes, and further combined with the visual output display taught by Percival, because Precival discloses a way to compare files as taught at column 4, lines 5 – 7 and the combination allows the changed performed to the code to be displayed comparing the difference before and after the optimization.

In regard to claim 12:

"generating an object code from an original source code;

optimizing the object code to produce an optimized object code;

decompiling the optimized object code to produce an optimized version of the source code, wherein the optimized version of the source code corresponds to an original version of the source code, modified to reflect an optimization in the optimized object code;"

Greyzck discloses generating an optimized code from an original source code (e.g., Figure 1; column 3, lines 25 – 29), but does not disclose that the change is performed in accordance with a compiler optimization. However, Brandes teaches the well-known procedure of decompilation in which source code is produced from optimized object code (column 1, lines 39 – 50). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the generation of optimized source code as taught by Brandes with source code optimization of Greyzck because the one would be motivated to combine the optimization techniques of Greyzck in a compiling function in order to avoid massive rewrites of old code as taught by Brandes at column 1, lines 9 – 31.

"simultaneously displaying the optimized source code and the original source code in separate windows of a user interface on an output device to visually indicate a change to the original source code as a result of the optimizing."

Neither Greyzck nor Brandes discloses simultaneously displaying the original source code and optimized source code on an output device to visually indicate a change to the original code. However, Percival discloses a visual display of file information indicating highlighted differences (column 4, lines 3 – 4; Figures 3 – 9) between two files. Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the generation of optimized source code as taught by the combination of Greyzck and Brandes, and further combined with the visual output display taught by Percival, because Percival discloses a way to compare files as taught at column 4, lines 5 – 7 and the combination allows the changed performed to the code to be displayed comparing the difference before and after the optimization.

In regard to claim 14, incorporating the rejection of claim 12:

"...wherein the original source code is compiled according to a compiler optimization comprising at least one of an inlining optimization, a common subexpression elimination, a loop invariant code removal, and a dead code elimination."

Greyzck compiles the optimized code having removed loop invariant expressions (see Abstract), but does not disclose generating an optimized source code wherein the original source code is compiled according to a compiler optimization. However, Brandes discloses generating source code from object code (column 1, lines 39 – 50). If the object code is optimized, then the recovered source code would be inherently optimized. Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine method of inputting the source code to be optimized of Greyzck with the source code generator of Brandes, because the

Art Unit: 2193

combination provides the capability to recover source code to thoroughly analyze the source code in order to rewrite or make changes (e.g., after optimization) as taught by Brandes at column 1, lines 9 – 31.

In regard to claim 15, incorporating the rejection of claim 12:

“...wherein simultaneously displaying comprises:

identifying a difference between the original source code and the optimized source code, where the difference is due to the compiler optimization on the original source code; and highlighting the differences on a user interface on the output device.”

Neither Greyzck nor Brandes discloses simultaneously displaying the original source code and optimized source code on an output device to visually indicate a change to the original code. However, Percival discloses a visual display of file information indicating highlighted differences (column 4, lines 3 – 4; Figures 3 – 9) between files. Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the generation of optimized source code as taught by the combination of Greyzck and Brandes, and further combined with the visual output display taught by Percival, because Percival discloses a way to compare files as taught at column 4, lines 5 – 7 and the combination allows the changed performed to the code to be displayed comparing the difference before and after the optimization.

In regard to claim 16 (an apparatus), rejected for the same reasons put forth in the rejection of claim 12 (a corresponding method).

In regard to claim 18 (an apparatus), incorporating the rejection of claim 16, rejected for the same reasons put forth in the rejection of claim 14 (a corresponding method).

Art Unit: 2193

In regard to claim 19 (an apparatus), incorporating the rejection of claim 16, rejected for the same reasons put forth in the rejection of claim 15 (a corresponding method).

In regard to claim 28 (a computer readable medium), incorporating the rejection of claim 20, rejected for the same reasons put forth in the rejection of claim 8 (a corresponding method).

In regard to claim 29 (a computer readable medium), incorporating the rejection of claim 20, rejected for the same reasons put forth in the rejection of claim 9 (a corresponding method).

In regard to claim 32 (a computer readable medium), rejected for the same reasons put forth in the rejection of claim 12 (a corresponding method).

In regard to claim 35 (a computer readable medium), incorporating the rejection of claim 32, rejected for the same reasons put forth in the rejection of claim 14 (a corresponding method).

In regard to claim 36 (a computer readable medium), incorporating the rejection of claim 32, rejected for the same reasons put forth in the rejection of claim 15 (a corresponding method).

10. Claims 2; 13; 17; 21, 22; 33, and 34 are rejected under 35 U.S.C. 103(a) as being unpatentable over Greyzck et al., U.S. Patent 5, 361, 354, in view of Brandes, U.S. Patent 5,946,484, as applied to claims 1 and 20 above, and further in view of IBM Technical Disclosure Bulletin NN9305305 (hereinafter referred to as IBM_TDB).

In regard to claim 2, incorporating the rejection of claim 1:

"...wherein the generating is performed in response to a request containing a type of the compiler optimization to be used by a compiler program when compiling the original source code."

Neither Greyzck nor Brandes discloses generating optimized code in response to a type of optimization to be applied to the source code. However, the IBM_TDB discloses user specified functions that run certain optimizations, while preventing other optimizations (page 1). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the generating of optimized source code as taught by Greyzck combined with Brandes, and further combined with requesting one type of optimization, while preventing another type of optimization as taught by the IBM_TDB, because the combination guarantees that certain variables are available for debugging that may otherwise not be available when the code is optimized as taught by the IBM_TDB at the top of page 3.

In regard to claim 13, incorporating the rejection of claim 12:

"...wherein generating is performed in response to a request containing a type of a compiler optimization to be applied on the original source code to generate the optimized source code."

Neither Greyzck nor Brandes discloses generating optimized code in response to a type of optimization to be applied to the source code. However, the IBM_TDB discloses user specified functions that run certain optimizations, while preventing other optimizations (page 1). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the generating of optimized source code as taught by Greyzck combined with Brandes, and further combined with requesting one type of optimization, while preventing

Art Unit: 2193

another type of optimization as taught by the IBM_TBD, because the combination guarantees that certain variables are available for debugging that may otherwise not be available when the code is optimized as taught by the IBM_TDB at the top of page 3.

In regard to claim 17 (an apparatus), incorporating the rejection of claim 16, rejected for the same reasons put forth in the rejection of claim 13 (a corresponding method).

In regard to claim 21 (a computer readable medium), incorporating the rejection of claim 20, rejected for the same reasons put forth in the rejection of claim 2 (a corresponding method).

In regard to claim 22 (a computer readable medium), incorporating the rejection of claim 20, rejected for the same reasons put forth in the rejection of claim 2 (a corresponding method).

In regard to claim 33 (a computer readable medium), incorporating the rejection of claim 32, rejected for the same reasons put forth in the rejection of claim 13 (a corresponding method).

In regard to claim 34 (a computer readable medium), incorporating the rejection of claim 32, rejected for the same reasons put forth in the rejection of claim 13 (a corresponding method).

11. Claims 5, 7; 25 and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Greyzck et al., U.S. Patent 5, 361, 354, in view of Brandes, U.S. Patent 5,946,484, and further in view of Mattson, Jr. et al., U.S. Patent 6,430,741 (hereinafter referred to as Mattson).

Art Unit: 2193

In regard to claim 5, incorporating the rejection of claim 4:

"...wherein displaying comprises:

displaying a number of times a procedure call in the original source code is inlined."

Neither Greyzck nor Brandes discloses a number of times a procedure call in the original source code is inlined. However, Mattson discloses method to analyze a computer program and displaying the number of times various data elements have been accessed (column 10, lines 25 – 36). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to modify the Greyzck invention to optimize the original source code, with the teaching of Mattson to display the number of time a specific function occurs (e.g., metering to determine the number of times a procedure call is inlined uses the same method as determining the number of times a data element is accesses). This combination provides a means, based on the visual display, to modify the code based on the result or to verify correctness as taught by Mattson at column 10, lines 46 – 49.

In regard to claim 7, incorporating the rejection of claim 6:

"...further comprising:

displaying a number of times a procedure in the original source code is executed by an interpreter of a visual machine program before being compiled by a run-time compiler of the visual machine program."

Neither Greyzck nor Brandes discloses a number of times a procedure call in the original source code is executed by an interpreter or a visual machine program. However, Mattson discloses method to analyze a computer program and displaying the number of times various data elements have been accessed (column 10, lines 25 – 36). Therefore, it would have been

Art Unit: 2193

obvious to one skilled in the art at the time the invention was made to modify the Greyzck invention to optimize the original source code, with the teaching of Mattson to display the number of time a specific function occurs (e.g., metering to determine the number of times a procedure call is inlined uses the same method as determining the number of times a data element is accesses). This combination provides a means, based on the visual display, to modify the code based on the result or to verify correctness as taught by Mattson at column 10, lines 46 – 49.

In regard to claim 25 (a computer readable medium), incorporating the rejection of claim 24, rejected for the same reasons put forth in the rejection of claim 5 (a corresponding method).

In regard to claim 27 (a computer readable medium), incorporating the rejection of claim 26, rejected for the same reasons put forth in the rejection of claim 7 (a corresponding method).

12. Claims 10, 11; 30 and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Greyzck et al., U.S. Patent 5, 361, 354, in view of Brandes, U.S. Patent 5,946,484, as applied to claims 1, and further in view of Shrader et al., U.S. Patent Application Publication US 2003/0005349.

In regard to claim 10, incorporating the rejection of claim 1; and **In regard to claim 11**, incorporating the rejection of claim 10:

"...wherein displaying comprises:

identifying a failed optimization on the original source code;

determining a reason for the failed optimization from a compiler used to optimize the source code; and

displaying the reason for the failed optimization."

"...wherein the identifying is performed in response to a user query."

Greyzck discloses the optimization of original source code before generating the object code (column 4, lines 3 – 33), but neither Greyzck nor Brandes discloses identifying a failed optimization, determining the reason, and displaying the reason. However, Shrader discloses an analysis of an application with detection of a failed function (an authentication), determines the reason for the failures, and displays the reason at the user's request (Abstract; also paragraph [0041]). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the generating of optimized source code as taught by combining Greyzck and Brandes, and further combined with the identification of failure, determining a reason for failure, and displaying the reason for failure as taught by Shrader, because the combination provides a means to determine the optimization failure and offers recovery suggestions as taught by Shrader at paragraph [0042].

In regard to claim 30 (a computer readable medium), incorporating the rejection of claim 20, rejected for the same reasons put forth in the rejection of claim 10 (a corresponding method).

In regard to claim 31 (a computer readable medium), incorporating the rejection of claim 20, rejected for the same reasons put forth in the rejection of claim 11 (a corresponding method).

Response to Arguments

13. Applicant's arguments filed 11/16/2004 have been fully considered but they are not persuasive:

The Applicant has argued:

Regarding claims 1, 3, 4, 6, 20, 23, 24 and 26, Greyzck, in view of Brandes fails to teach or suggest generating, from optimized object code, the optimized version of source code, wherein the optimized version of source code corresponds to an original version of the source code, modified to reflect an optimization in the optimized object code. Specifically, the techniques of Brandes fail to generate object code that has a relationship to an original version of the source code. Brandes discloses generating source code for common patterns of object code. Brandes, Abstract. *However, the relationship between source code and object code is not one-to-one. Thus, many different versions of source code will compile to an identical sequence of machine instructions (i.e., object code). As a result, decompiling object code does not produce the original source that was used to generate the given object code.* In other words, the source code generated by the techniques taught by Brandes have no relationship to the original source code, in fact Brandes is directed to situations where the original source is lost, or otherwise unavailable. Therefore, the source code generated using the techniques disclosed in Brandes is not likely to bear any correspondence with the original source code. (Emphasis added)

Thus, Brandes (when combined with Greyzck) fails to teach or suggest generating, from optimized object code, the optimized version of source code, wherein the optimized version of source code corresponds to an original version of the source code, modified to reflect an optimization in the optimized object code.

Examiner's response:

It appears that the Applicant is arguing that a one-to-one relationship between source code and object code exists in the extant application such that if the object code is decompiled, then an identical source code is reconstructed or reproduced except for sections of the code that were optimized. If that is in fact the intended patentable feature, then that feature is not properly expressed in either the claims or the specification.

For example, simply characterizing the decompiled code as “corresponds to an original version of the source code” or having “a relationship to an original version of the source code” does not capture the apparent intention or concept of this proposed invention as expressed in the emphasized section of the Applicant’s argument above (*However, the relationship between source code and object code is not one-to-one. Thus, many different versions of source code will compile to an identical sequence of machine instructions (i.e., object code). As a result, decompiling object code does not produce the original source that was used to generate the given object code*). The decompiled code in the applied art as cited in the office action would indeed have a “relationship” to, and even a “correspondence” to, in a broad sense of the term, the original source code even though it would not be *identical* to, or reproduce a *one-to-one* relationship with the parts of the source code that were not optimized. The amended claims recite *“generating, from optimized object code, the optimized version of source code, wherein the optimized version of source code corresponds to an original version of the source code, modified to reflect an optimization in the optimized code*. It would be reasonable to expect that any de-optimized code corresponds to, or represents, or has a relationship to the original source code since one would expect the same results from the execution of the resultant optimized executable code. The optimized source code must be related to the original source code since the reconstructed code is ultimately derived from the original source code.

Additionally, the specification provides no guidance or clarification. Nothing in the specification supports the claim that the optimized version of the source code “corresponds to an original version of the source code,” as the amended claims now recite. Paragraph [0042] comes closest: “The compiler 116 then optimizes the source code 202 to generate object code 204 in

Art Unit: 2193

accordance to the compiler optimization selected in the request. The decompiler 130 then converts the object code 204 into an optimized source code 214. The optimized source code 214 represents the original source code 202 with the selected compiler optimization implemented therein. Once the optimized source code 214 is generated, the editor 114 may display the optimized source code 214 and/or the original source code 202. In one embodiment, the editor 114 may display both the original source code 202 and the optimized source code 214 with the changes highlighted or otherwise marked.” (Emphasis added). The applied art, as shown above, “represents” the original source code, albeit not an identical representation, but an identical representation is not supported by the specification as shown in paragraph [0042]. The specification does not convey the idea that the optimized source code is a one-to-one replica, or exactly corresponds to an original version of the source code, except for changes in the optimized parts, as seems to be suggested in the arguments of the current amendment.

Conclusion

14. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Lawrence Shrader whose telephone number is (571) 272-3734. The examiner can normally be reached on M-F 08:00-16:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner’s supervisor, Kakali Chaki can be reached on (571) 272-3719. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Art Unit: 2193

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Lawrence Shrader
Examiner
Art Unit 2193

4 August 2005



ANIL KHATRI
PRIMARY EXAMINER